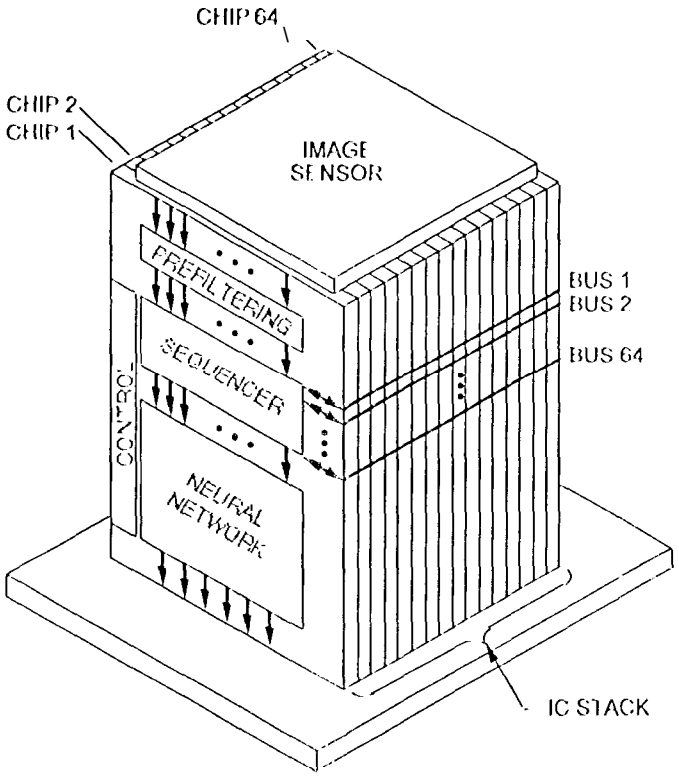
64x64 Analog Input Array for 3-Dimensional Neural Network Processor

T. A. Duong, 'J'. Thomas, "1'. Daud, A. Thakoor, and B. Lee[1]

Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109
1 Irvine Sensors Corporation
Irvine, CA 92626

I. Introduction:

In pattern recognition and classification for spatio-temporal problems, one of the most challenging tasks is to provide a good and valid solution in real-time. Because of time constraints, sclflw'are-based neural network approaches may not be suitable for practical usc. Hardware solutions seem to be good candidates for this class of problems. Currently, the three Dimensional Analog Neural Network (3-1)ANN) [1] is an efficient approach to solve spatio-temporal problems in three dimensional hardware..



The 3-DANN[1] consists of an IR focal plane, a 3-Dimensional Neural Conditioning Module (3-DNCM) block, and a 3-Dimensional Neural Processing Module (3-1)NPM) block (see Figure 1). The IR focal plane receives a contiguous image, and is operated at low temperature (77K) in order to reduce noise. The fall 64x64 analog, image output of the IR focal plane is fed to the 3-DNCM which behaves as a spatial filter, filtering all unnecessary infer-rna[iorl. The output of the 3-DNCM is a fully parallel 64x64 analog array providing a cleaner image to the next stage. The 3-DNPM receives this image for further processing to obtain a desired solution.

The 3-DNPM contains 64 stacked chips which form a cube. Each chip has a 64x64 array of 8-bit synapses, and is able to perform a fully parallel multiplication between 64 analog voltage inputs and 64 columns of the synaptic ar ray (each column contains 64 synapses). In addition, each column is summed together to obtain 64 summation lines from each chip. Furthermore, these lines are connected with their counterparts from the other chips

to provide 64 final current outputs. The 64 corresponding columns moss sill 64 chips combine to form 64 templates (e.g. column 1 of all 64 chips stores template 1, c t c.. .). Therefore, the 64 final current outputs represent the inner products of the 64x64 input array and the 64 available templates.

1I. New 3-DANN-M Approach

For the current project, the IR focal plain and 3-DNCM arc removed and replaced by a single chip that lies cm top of the cube. This approach is called "3-I) ANN-M". The arriving image is no longer from an IR focal plane, but it is rather from a 256x256 digital image which is prefiltered and stored in a memory block (Figure 2).

in order to exploit the full capability and computational power of the 3-DNPM cube, the input chip, called the "Column Loading Input Chip (CLIC)" must provide a 64x64 array of 8-bit analog inputs in every 250 ns time frame [2].

An immediate application of the CLIC/3-DNPM system is known as tbc "Vigilante" project, the details of which are not discussed in this paper. However, one of the requirements is to obtain a sub-window (64x64) from a 256x256 image and send this sub-image to the 3-DNPM for further processing (Figure 2).

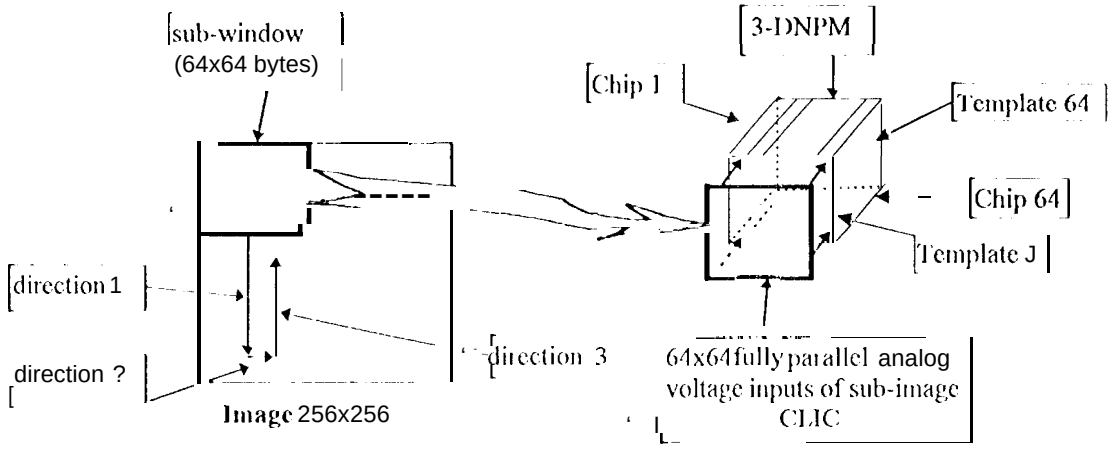


Figure 2: Sub-image selection. In this figure, a sub-image (64x64) is drawn from the 256x256 image. The sub-image is rastered one row down at a time (direction 1), cmc column right (in direction ?), and rastered one row up at a time (in direction 3).

in this paper, we only consider the CLIC chip which receives a digital sub-window (64x64 bytes) and conver Is it into an analog voltage array. The sub-image is rastered in the or iginal image one column/row at a time within ?50 ns, and the convulsion from the 1)igital-to-Analog Converter (D AC) array requires less than 1 SO ns. In other words, every ?S0 ns, a wl iole new column/row (64 bytes) is updated to the sub-image, less than 150 ns is required for the digital to an analog conversion, and J 00 ns is available for the 3-DNPM to do further in formation processing. In addition, the sub-i mage earl be rastered down/up one row at a time or r ight one row without losing the overhead time for reloading the new sub-image. Furthermore, the Cl .IC has a 64x64 local voltage output array which is available on 4,096 metal3 pads, 66x66 μm^2 in size. Each cell size is 101 .6x 101.6 μm^2, and the complete design is done in a 0.8μm HP CMOS process.

III. Technical Approach:

In Figure 2, a 64x64-byte sub-image is loaded from the original (256x256) image in the digital domain. "1'here are several important features to the CLIC:

- the overhead time, which is required to fill in the first 64x64-byte array in digital form, is ignored.
- every column/row (64 bytes information) rc.quilts 250 ns to completely load up/down or right into the CLIC after the first filled-in 64x64 array.
- the 64 bytes arc loaded in row up/down, or in column right (see figure 2 with explanation).

- the 4096 DAC in the CLIC provides fully parallel inputs to 3-DANN after 150 ns settling time with 8-bit precision.

The remaining 100ns is used for 3-DANN to process the information before new analog window information is available from the CLIC.

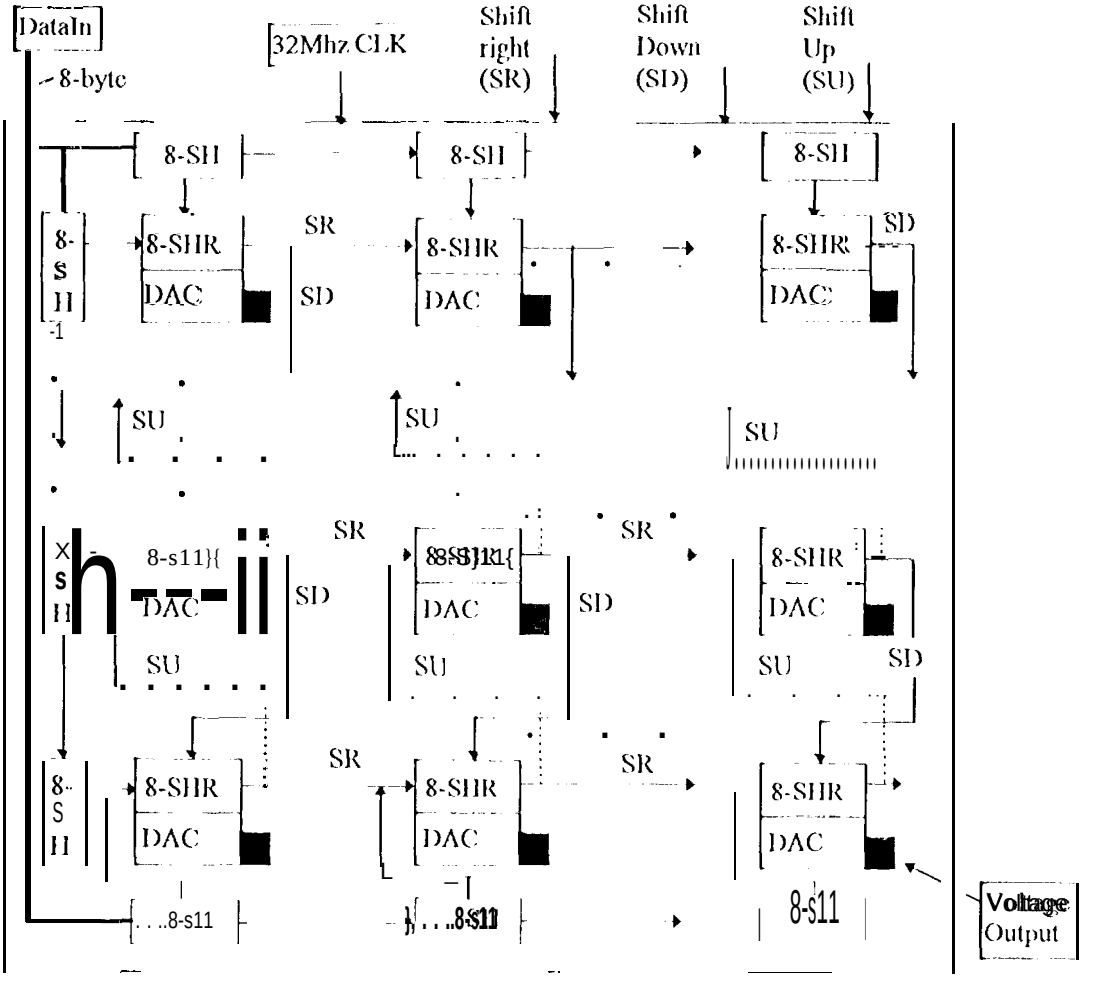


Figure 3: Architecture of CLIC input chip. This chip contains 64x64 MDACs that can shift a whole array right or down/up. The right most column or the bottom row will be shifted out when shift-right or shift-down is selected, respectively). The SRAM holds an 8-bit byte for each pixel in the sub-image. The solid squares arc the analog voltage outputs.

The CLIC architecture that is shown in Figure 3 contains:

1. A 64-bit digital input which is run with a 32-MHz clock to load 64 bytes within 250 ns.
2. Three independent controls and non-overlapped select lines that enable a column to shift op or down, or a row right.
3. Thu ee sets of 64-byte shift registers (shift-up, shift-down, and shift-right) which have 8.byte parallel-in/8-byte parallel-oot so that 64 bytes can be completely shifted in one of three directions.
4. MDACs arranged in a 64x64 array. Each MDAC has an 8-bit parallel-in/i>arallcl-out SRAM-SHIFT and 8-bit fully parallel DAC converter. The 8-bit SRAM-SHIFT can hold 8-bit information like a normal SRAM, and it also is able. to transfer 8-bit information as an 8-bit parallel in/parallel out shift register (see the design section).

1)AC receives in parallel the 8-bit complementar y data to convert it into an analog voltage with a reference voltage of 3.5 V.

E.g., the data with all bits ON will set all bits OFF in the DAC in which case the voltage output remains at 3.5 volts, and vice versa.

a) **SRAM-SHIFT design:**

SRAM-SHIFT (see Figure 4) is required to hold 8 bits in an SRAM and be able to Shift-in or Shift-out in a fully parallel fashion. This design can be done as a normal latch; however, the space constraint for a small sired cell (101 .6x101.6 μm^2) could not be achieved using a normal latch approach.

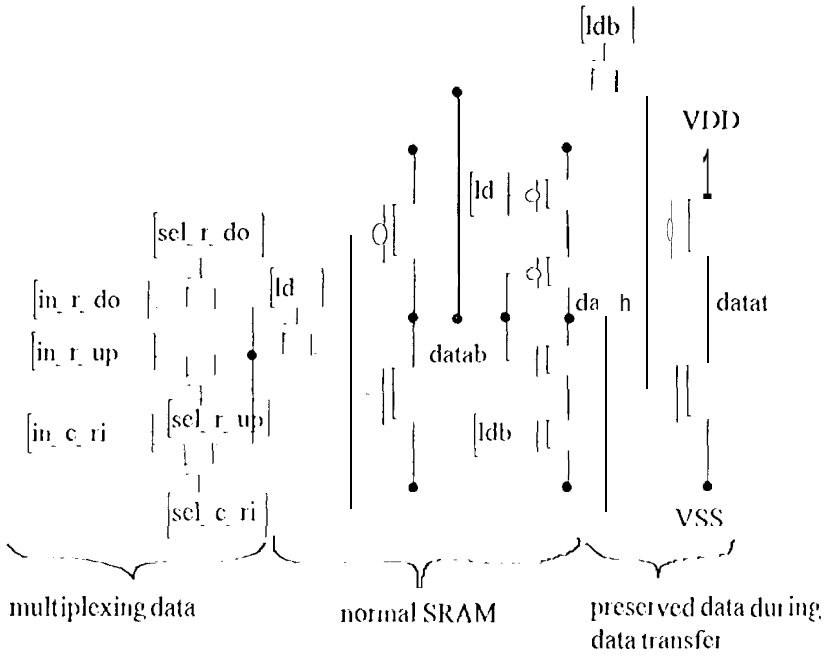


Figure 4: SRAM-SHIFT cell. In this figure, the SRAM-SHIFT contains three blocks: multiplexing data, SRAM, and preserved data for transferring without any distortion.

The SRAM-SHIFT design is explained as follows:

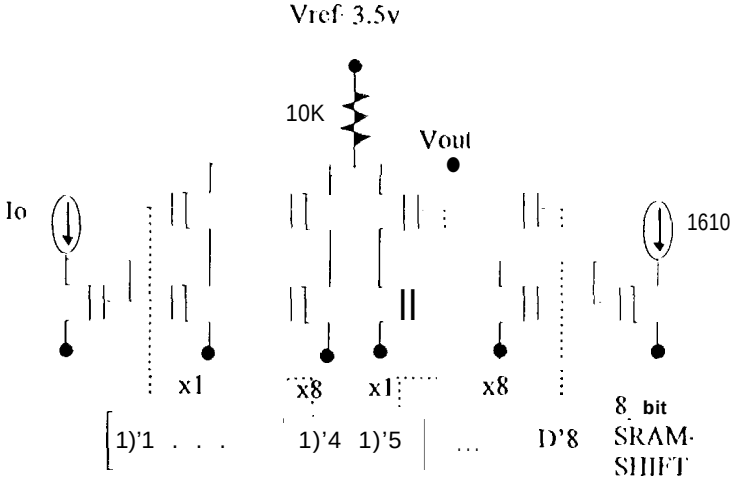
- The multiplexing data block allows us to select a source of data to use, which can be from the previous output from the immediate upper cell / lower cell, or from the cell to the immediate left. All the control select lines are common for the whole chip, but the input (in_r_do, in_r_up, anti in_c_ri are locally connected)
- The SRAM is a normal SRAM. The *ld* signal (see Figure 5) ON enables writing to the SRAM cells, and OFF holds the data, whatever is written. During the writing phase, the previous data *datah* is destroyed. Therefore, a preserved data block is introduced to preserved data *datat*
- The preserved data block is isolated when the *ld* signal is ON and can completely transfer the preserved data to the next cell without any distortion to that cell.

The combination of these three blocks will provide us with a robust design to hold and shift data within the given space constraint of the 3-DANN design.

b) **DAC converter:**

The data is stored in a SRAM-SHIFT ccl]. The complementary data is used to convert the analog, voltage. The conversion is performed as follows:

$$Vout = 3.5 - 10000 \{\$' I.(1 - D_i)2^{i-1} + \sum_{i=5}^{8} 16I_0(1 - Di)2^{i-1}\} \quad (1)$$



* D'i denotes the complementary output Of bit i from SRAM-SHIFT

Figure 4: DAC converter. In this figure, the complementary digital output (D'i) is converted into current, then converted to analog voltage through a 10KΩ resistor. E.g., When the digital value is zero, all bits in the DAC are "ON", and the output voltage is pulled down to 2.5 V.

The global currents I_0 and 161_0 are injected from outside of the chip so that matching between them is well-controlled. A 10KΩ resistor is provided for each MDAC, obtained through the resistance of an Nwell. When all the data bits are ON, the second term on the right side of equation (1) is zero, the voltage output is 3.5 V, data bits are OFF, and 255 units of current source I_0 are drawn from 3.5 V to lower the voltage level to the desired value (nominal ?.5 V). Because the conversion is done locally, the parasitic capacitance. is small, and the speed obtained is very fast (1s01,s).

IV Simulation results:

In this simulation, there are two critical aspects: settling time and linearity of the 8-bit DAC:

a) Settling time of DAC:

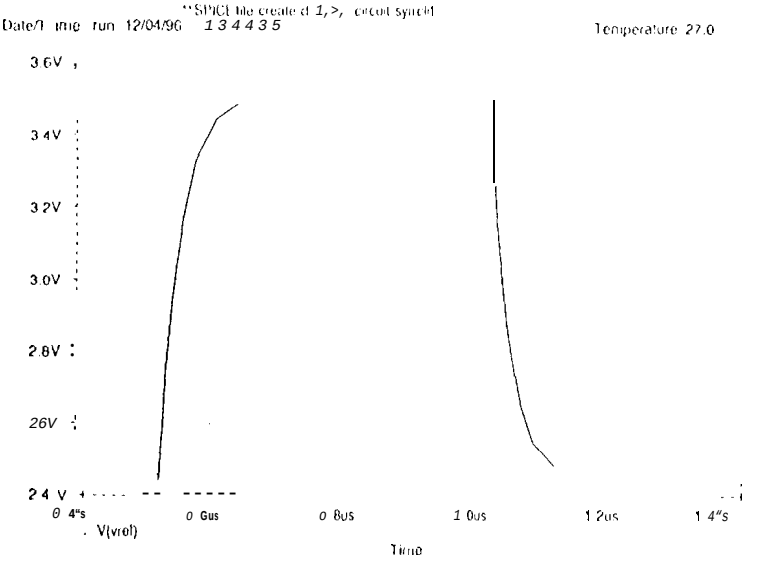


Figure S: Conversion time of the DAC. About 150 ns are required to convert the digital value to the output analog voltage with 8-bit precision.

in figure 5, the simulation shows that the settling time for DAC is 150 ns after the data is shifted to the SRAM-SHIFT.

b) Linearity of DAC:

Figure 9 shows the DC simulation results when each bit is turned on onc at a time to obtain all the combinations of the possible 8-bit levels:

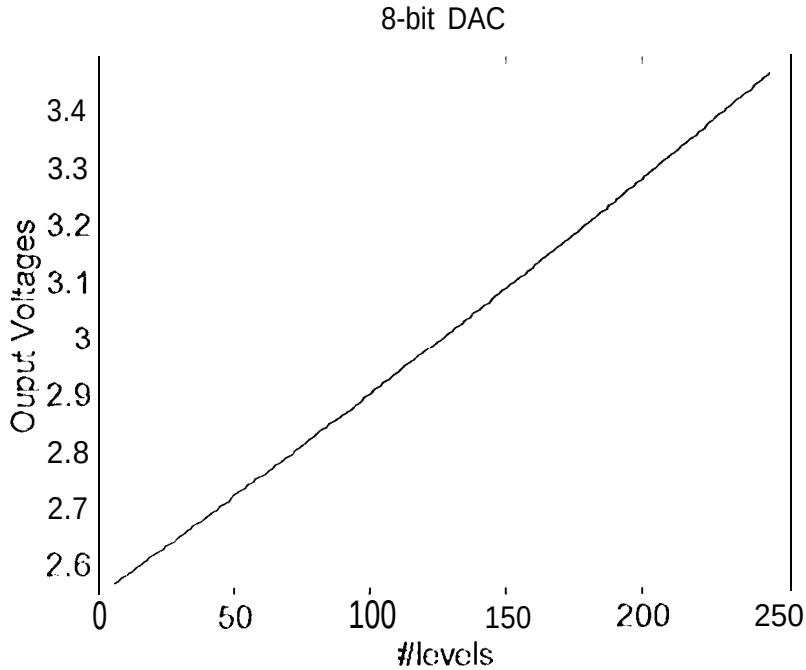


Figure 6: 8-bit Digital to Analog, Converter (8-bit DAC). The x-axis rc.presents 8-bit digital values and the y-axis represents the output analog voltages.

V Application:

For spatio-temporal classification problems, 3-D ANN-M is designed to search for the predefined objects in a fail 256x256 image.. Y is a 64x64 input image which is drawn and rastered for the entire or igiiial image to classify the objects in that image.

Let Y be a normalized input image (64x64 input array) to the 3-DNPM, which can be viewed as a 4096x 1 vector in one dimension for convenient formulation. $T_1 \ldots T_{64}$ arc the normalized, identified templates which arc stored in each planar column array of 3-DNPM, and $T_1 \ldots T_{64}$ are aiso viewed as 4096x i vectors.

Every 25011s, the computation of the 3-DNPM will provide 64 current outputs as described below:

$$O(64x1) = \begin{vmatrix} T_1^T(1x4096) \\ . \\ T_{64}^T(1x4096) \end{vmatrix} Y(4096x1) = \begin{vmatrix} \cos(\theta_1) \\ \\ \cos(\theta_{64}), \end{vmatrix} \qquad (2)$$

with θ_i the angle between (Y, T_i)

With the combination of the CLIC chip and the 3-DNPM cube, we have a Dew' machine which is capable of several approaches to solving spatio-temporal classification problems in real time. For the time being, two approaches are being used:

1. In the first approach (the straightforward approach), the 3-DNPM cube contains 64 searched templates. Each template is a 64x64 pixel anay stored perpendicularly to the chips across the columns e.g., cacb column j of each chip will belong to template j (see Figure 2). In equation (2), the inner product between a 64x64 sub-image Y and 64 templates T_1-T_{64} can be done within 250ns and a 64-input Winner-Take-All chip will determine the best match between the input sub-image Y and the template for classification. A requirement for this approach is to have Y and Ts' normalized. The sub-image Y is rastered one column/row at a time to look for the best match through the whole 256x256 image array. Finally the best match between the sub-image and the template is again evaluated by their overlapping level to determine the confidence decision. The conclusion is drawn based upon this confident level whether the object is identified or not.
2. In the second approach, the templates T_1-T_{64} arc eigen vectors of the scare.bcd targets which arc mainly obtained through the principle components appr oach[3]. The 64-outputs of the inner products between Y and "1'1"1'61 arc. fed to a multi-layer perceptron learning network [4] for classifying the objects. The results obtained in software simulations for current objects arc optimistic so far[3].

VI Conclusion:

In this paper, wc demonstrated that in SPICE, the CLIC has the capabilities to overr ide the bottleneck of the 64x64 input array in which 3-DNPM can be exploited at the full speed of the most powerful processors. With the combination of CLIC and 3-DNPM cube, the speed of such machines can be available to solve spat io-temporal problems in r eal time.

Acknowledgments:

The research described herein was performed *by the* Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly spon sored by the Ballistic Missile Defense Organization/Innovative Science and Technology Office (BMDO/IST), the Office of Naval Research (ONR), the Advanced Rest.arch Projects Agency (ARPA), and the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs. Steve Sudda, Udomkesmalee Suraphol for their supports.

References:

1. T. A.Duong, et al. "Analog 3-D Neuroprocessor for Fast Frame Focal Plane Image processing," *Simulation Journal*, Vol. 65, No.*1*, pp. *11-25*, July 1995.
2. T. A.Duong, et al. "1 .ow Power Analog Neurosynapse Chips for a 3-D "Sugar cube" Neur oprocessor," *Proc. Of IEEE Int'l Conf. On Neural Networks (ICNN/WCNN)*, Vol. 3, pp. 1907-1911, June 28-July 2, 1994, Orlando, Florida.
3. C. Paggett, "Detection and orientation classifier for the VIGILANTE image processing system", **submitted to** SPIE Con ference in Or lando, April 1997.
4. D.E. Rumelhart, and J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vd. I: Foundation.* MIT Press, Cambr idge, M A i 986.